CENTRE FOR
SYSTEMS
SCIENCE

LCCR

Simon Fraser University, Burnaby, B.C., Canada V5A 1S6 / (604) 291-4704

Estimating Testing Effectiveness of the Circular Self-Test Path Technique

Slawomir Pilarski, Andrzej Krasniewski and Tiko Kameda

CSS/LCCR TR 90-10

Revised September 1990

92-05795/

Estimating Testing Effectiveness of the Circular Self-Test Path Technique*

Slawomir Pilarski†

Andrzej Krasniewski††

Tiko Kameda†

Abstract

In this paper we study the effectiveness of a random built-in self-test technique for VLSI circuits. This technique, called the circular self-test path, is applicable to circuits that consist of combinational blocks and registers. In particular, the effectiveness of test pattern generation, the effectiveness of test response compaction, and fault coverage are examined. The test generation effectiveness is evaluated by the fraction of all possible test patterns applied during a testing session to the circuit under test. The compaction effectiveness of the CSTP technique is measured by the probability of aliasing, and fault coverage by the fraction of all permanent faults that are detected. For all these measures, simple formulae are developed. They give very accurate estimations without detailed circuit simulation. To demonstrate their accuracy, the estimates obtained by the formulae are compared to the results obtained by extensive simulation experiments.

---

* This work was supported in part by the Advanced Systems Institute of British Columbia, and in part by the Natural Sciences and Engineering Research Council of Canada.

† School of Computing Science, Simon Fraser University, Burnaby, B.C., Canada V5A 1S6. Netaddress: {pilarski, tiko}@cs.sfu.ca.

†† Warsaw University of Technology, Institute of Telecommunications, Nowowiejska 15/19, 00-665 Warszawa, Poland.

1. INTRODUCTION 2

2. TESTING EFFECTIVENESS 5

2.1. PREVIOUS RESULTS 6

2.2. ASSUMPTION ON REGISTER ADJACENCY 7

3. EXPECTED STATE COVERAGE 7

3.1. DEFINITION AND PROPERTIES 8

3.2. APPROXIMATION FORMULA 10

3.3. SIMULATION STUDY USING A PROBABILISTIC MODEL 12

3.3.1. PREVIOUS MODEL 12

3.3.2. NEW MODEL 13

3.3.3. RESULTS 13

3.4. SIMULATION USING SAMPLE CIRCUITS 17

4. THE LEAST PROBABLE STATE 19

5. PROBABILITY OF ALIASING 21

6. FAULT COVERAGE ESTIMATION 30

7. CONCLUSIONS 33

8. APPENDIX 34

REFERENCES 38

1. INTRODUCTION

In order to test VLSI circuits effectively, testability has to be considered at the circuit design stage [WIPA82, BMS87]. Among various strategies for designing easily testable circuits, self-testing appears to be the most effective [VAB84, DTM90]. The continually growing complexity of VLSI circuits and interest in low volume application specific integrated circuits make built-in self-test (BIST) techniques especially attractive. BIST reduces the complexity of automatic test equipment by moving some or all external tester functions onto the chip itself. The penalty paid for BIST is extra silicon area and circuit speed degradation.

The implementation of BIST usually requires selected blocks of a VLSI chip to be replaced with more complex ones. For example, built-in logic block observation techniques [KMZ79] replace regular registers with multifunctional registers. The functionality of all BIST structures is transparent during the normal system operation. During a testing session the chip is partitioned into a set of subcircuits, and several such subcircuits are then tested simultaneously. Usually, several testing sessions are necessary to test the whole chip. A built-in circuit, called the **test pattern generator**, generates and applies test patterns to the inputs of the circuit under test (CUT), while another built-in circuit called the **response compactor** compacts the set of CUT responses into a single **signature**. This signature constitutes the basis for pass/fail analysis.

Most BIST techniques implemented in modern VLSI circuits, including the latest generation of microprocessors and application specific integrated circuits, use linear feedback shift registers (LFSRs) for test pattern generation and LFSR-based parallel signature analyzers for response compaction. Unfortunately, the silicon area consumed by these structures and their relatively complex controllers seriously limits the applicability of BIST in practice.

An alternative BIST technique, called the circular self-test path (CSTP), sometimes also referred to as circular BIST [POLB88], has recently been developed [KRPI87, PIKR88, KRPI89]. This random testing technique, although developed independently, is similar to the simultaneous self-test concept presented in [BAMC82]. The CSTP is applicable to circuits that consist of combinational logic blocks and registers. The idea behind CSTP is to arrange selected registers into a

circular path, as illustrated in Fig. 1, where *CL* denotes a combinational block and *reg* denotes a register. Shaded registers constitute the circular path.

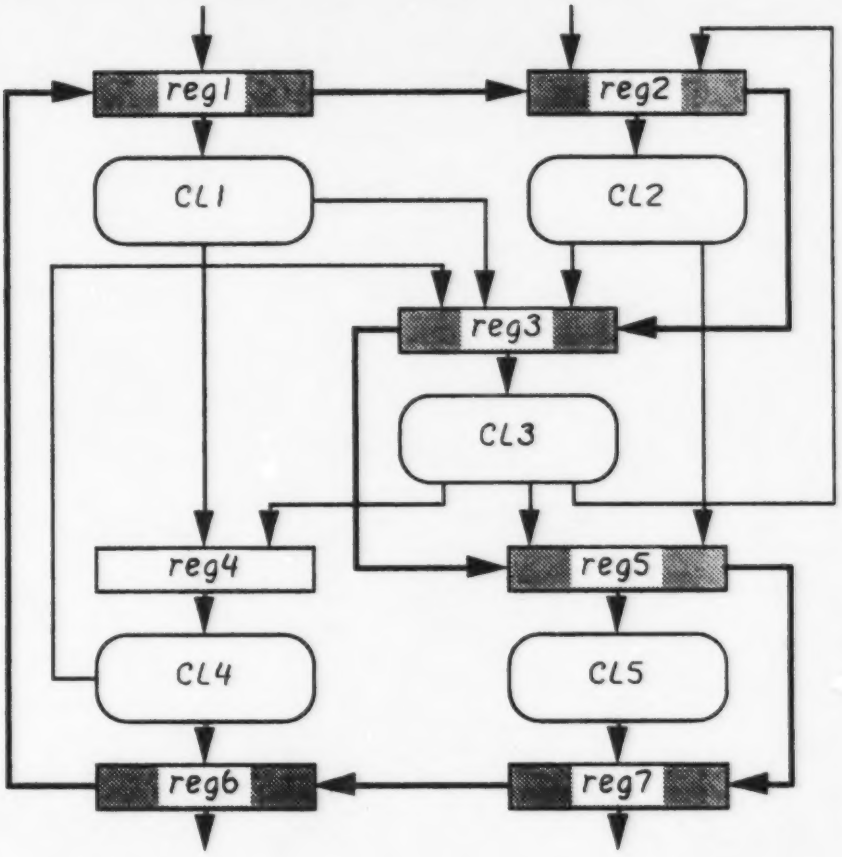

Fig. 1. A circular self-test path.

Each CSTP register consists of a number of cells, where a cell contains a flip-flop, an EX-OR gate, and a circuit selecting the mode of operation. All cells form a loop, so that each cell has one preceding and one succeeding cell. The order of flip-flops in the CSTP in Fig. 1 is indicated by the thick arrows. During the normal mode of chip operation, the cells of the circular path serve as conventional D-type flip-flops. In the test mode, however, the modulo 2 sum of the regular input and the state of the preceding cell is written into the flip-flop of each cell. Thus, the circular path provides the CUTs with test patterns and, simultaneously, compacts the circuit responses (Fig.2). A pass/fail decision is made by comparing the signature, i.e., the stream of

bits that appear at a preselected point of the circular path within a specified time interval at the end of a testing session,[1] with that expected from a fault-free CUT.

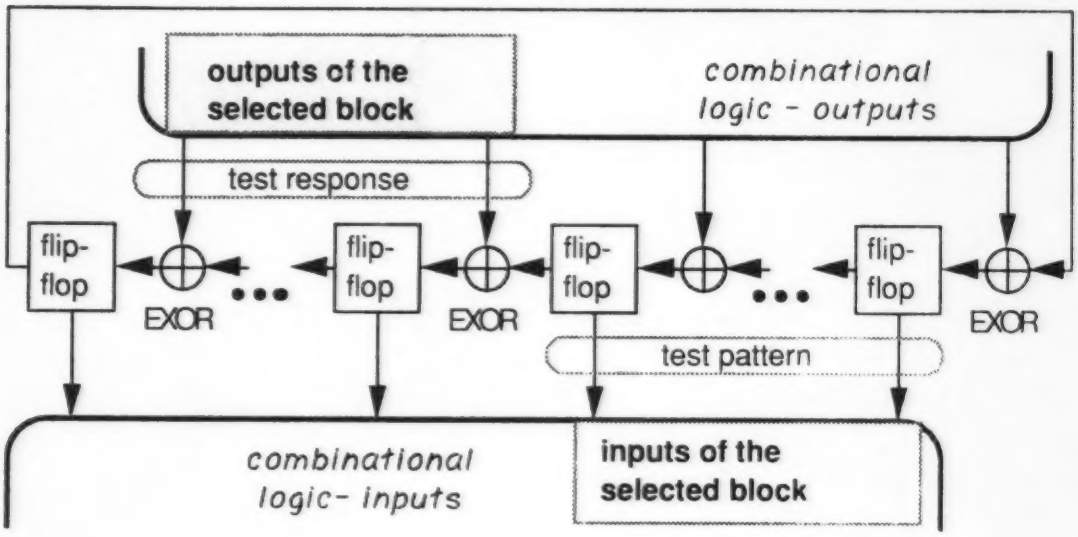

Fig. 2. A circular self-test path in the test mode.

The CSTP technique has been shown to offer a number of attractive features [KRPI89]. In particular:

- The silicon area overhead is substantially lower than that required by the earlier BIST techniques.
- The whole circuit is tested in one testing session, thereby minimizing the complexity of the test controller.
- As the circular path may include an arbitrarily selected set of circuit registers, the circuitry overhead can be traded-off with the test quality.
- The CSTP technique lends itself very well to design automation.

The major concern one may have about the CSTP technique is the quality of testing it offers. This paper addresses this particular problem in-depth. More specifically, the

---

[1] An alternative way is to read the state of a preselected set of CSTP flip-flops at the end of a testing session [STRO88a].

effectiveness of test pattern generation, the effectiveness of response compaction, and the fault coverage are studied. The paper is organized as follows. Section 2 summarizes some results on testing effectiveness published in [KRPI89] and states an important assumption under which these results were derived. In section 3.1, we define the notion of state coverage used to evaluate test pattern generation effectiveness, and present a theorem linking expected state coverage with the probabilities of generating test patterns. This theorem serves as a basis for developing useful formulae for estimating expected state coverage in subsequent sections. Section 3.3 presents a new probabilistic model which is used for circuit simulation and for deriving estimates for expected state coverage. Section 3.4 presents some statistical data on expected state coverage obtained from the simulation of a large number of circuits belonging to a particular circuit class. Those results confirm the validity of the circuit model presented in section 3.3. Section 4 introduces a formula for estimating the probability that a test pattern is covered during a testing session. This probability is a measure of test generation effectiveness. Section 5 is devoted to the analysis of aliasing. It presents results of extensive circuit simulations and a simple formula that estimates the probability of aliasing. Section 6 shows the relation between expected state coverage, the probability of aliasing, and expected fault coverage. Finally, section 7 concludes the paper. The appendix contains a proof of the formula presented in Section 4.

2. TESTING EFFECTIVENESS

The quality of testing is often judged by **fault coverage**, i.e., the percentage of all faults in a particular class that a given test procedure can detect [MCCL89]. The most common class of faults considered are single stuck-at faults. Sometimes this class is confined to detectable faults only. This extra restriction simplifies the comparison of the test quality achieved by different testing procedures.

In the case of BIST, the quality of testing for permanent detectable faults is determined by two factors: the effectiveness of random test pattern generation and the effectiveness of test response compaction. For a combinational block, the effectiveness of random test pattern generation can be expressed as the fraction of all possible input patterns that are applied to the inputs of the block under test during a testing session. The effectiveness of test response compaction is usually evaluated by the probability of aliasing (error escape), i.e., the probability that the test response

compactor produces a fault-free signature for an erroneous response sequence received from the CUT.

2.1. PREVIOUS RESULTS

Using the CSTP technique, the probability of aliasing can be made arbitrarily low by examining increasingly longer signatures [KRPI89]. The effectiveness of test pattern generation was, therefore, the primary concern. The problem of evaluating the effectiveness of test pattern generation, recognized as difficult for random testing [SABA84], becomes even more difficult for circular paths. This is because the test patterns produced by a circular path depend on the functionality of the CUT. A comprehensive simulation study was performed to examine the test generation capabilities of circular paths [KRPI87, KRPI89], leading to the following conclusions:

- Shortly after a testing session is started, the probability that the output of any flip-flop in a circular path takes value 1 converges to 0.5 regardless of the circuit characteristics and the initial state of the path.[2]
- The effectiveness of test pattern generation is almost independent of the architectural and functional characteristics of the CUT.
- CSTP provides nearly exhaustive test capabilities - 99% of all possible vectors are generated within a period a few (4-8) times longer than that required by a LFSR.

The second conclusion is, in our opinion, the most valuable and the most surprising result of the simulation study performed. Based on this conclusion, one can estimate the effectiveness of test pattern generation by a circular path, assuming that for any k-bit section of the path all 2^k possible vectors are equally likely at any time instant and the probabilities of any two vectors are independent. Under this assumption, a procedure for calculating the expected number of different patterns generated was presented in [KRPI87]. Unfortunately, this procedure has turned out to be computationally very expensive.

---

[2]The ratios of zeros to ones for some real circuits are presented in [STRO88a][STRO88b]

2.2. ASSUMPTION ON REGISTER ADJACENCY

Although it was not stressed in the previous papers [KRPI87, KRPI89] the conclusions presented in Section 2.1 were based on the assumption that there is no statistical correlation between the streams of bits applied to the two inputs of any of the CSTP EX-OR gates (see Fig.2). In practice, it means that there is no **register adjacency** [HUPE87, STRO88b], i.e., the value of the test response bit fed to a CSTP cell does not depend on the state of the preceding flip-flop.

Register adjacency, in general, affects the probability that the output of a flip-flop in a circular path takes value 1. In practice, this probability could be very close to 0 or 1. In fact, it is not difficult to imagine a circuit, where a CSTP flip-flop always takes value 1. In such a case, the conclusions presented in Section 2.1 are not valid. Obviously, some test patterns will never be produced. Similarly, only a few faults may make the flip-flop take value 0. This means that the aliasing probability is close to 1, if the signature is collected at the flip-flop's output. Register adjacency should, therefore, be avoided in practice. Sometimes this can be done by choosing the order of flip-flops that are to be included in CSTP. For some circuits, however, this method may not work. In such cases, extra flip-flops may be used. They should be "transparent" during the normal circuit operation and used to break register adjacency in the test mode.

It is assumed in this paper that all circuits that we consider are free of register adjacency.

3. EXPECTED STATE COVERAGE

For a given circuit, a given initial state of its registers, and a given testing session length, one can determine the number of distinct test patterns applied to each circuit block. To do that, however, one needs to simulate the circuit behavior, which may be expensive especially for very large circuits. Such simulation requires a detailed circuit model and is invalidated by any change made to the circuit. An inexpensive method to estimate test generation effectiveness would, therefore, be very useful. In this section, we use probabilistic models of circuit behavior to develop a method which estimates the expected number of different test patterns generated during a testing session. This method will be preferable to the lengthy procedure proposed in [KRPI87].

3.1. DEFINITION AND PROPERTIES

Consider a k-input combinational block driven by a set, A, of circular path flip-flops,[3] as illustrated in Fig.3.

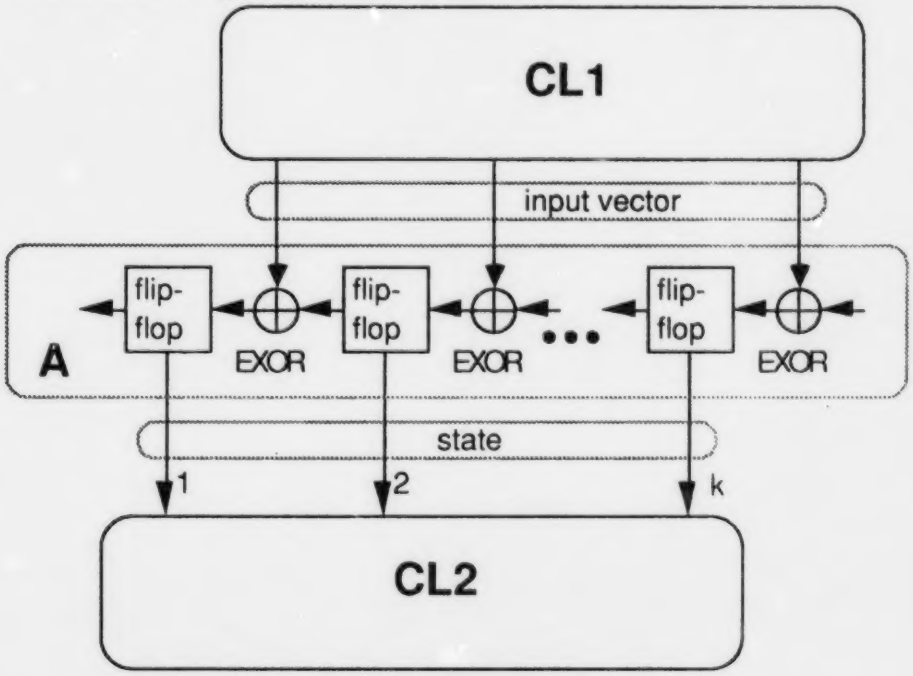

Fig. 3. CSTP section.

The flip-flops' output values constitute an input vector for combinational block CL2, and are referred to as the **state** of A. We judge the quality of testing by the number of different states that occur during a testing session. Thus, we are interested in **state coverage**, which, in our terminology, means the fraction of all 2^k possible states that are actually generated (or **covered**) during a testing session. As mentioned before, a probabilistic model of circuit behavior will be used to study the expected value of state coverage. Before we derive a formula that estimates it, we formulate and prove a general theorem that shows a relationship between **expected state coverage** (ESC, for short), defined below, and the probabilities that different states are covered during a testing session.

---

[3]The flip-flops need not to be adjacent in the circular path.

We use the following notation:

$p_{A,s}(T)$: probability that state s, $s = 0,1, \ldots, 2^k-1$ (binary vectors are represented by the corresponding numbers), is covered during a testing session of T clock cycles at the outputs of A.

$ESC_A(T)$: expected state coverage for A and a testing session of T clock cycles. More formally, let SEQ denote the set of all possible sequences of states that may occur at the outputs of A during testing sessions of T clock cycles. Then

$$ESC_A(T) = 2^{-k} \cdot \sum_{seq \in SEQ} N(seq) \cdot p_A(seq) , \qquad (3.1.1)$$

where N(seq) is the number of different states covered by sequence seq, and $p_A(seq)$ is the probability of sequence seq appearing at the outputs of A.

Theorem 1 [PIKR88]:

The expected state coverage for a set A of k circular path flip-flops is equal to the average probability of covering a state, i.e.,

$$ESC_A(T) = 2^{-k} \cdot \sum_{s=0}^{2^k-1} p_{A,s}(T).$$

Proof: From basic probability theory, it follows that

$$p_{A,s}(T) = \sum_{seq \in SEQ} p(s<<seq) \cdot p_A(seq), \qquad (3.1.2)$$

where p(s<<seq) denotes the probability that the sequence seq covers state s (clearly, p(s<<seq) is either 0 or 1). Thus,

$$\sum_{s=0}^{2^k-1} p_{A,s}(T) = \sum_{s=0}^{2^k-1} \sum_{seq \in SEQ} p(s<<seq) \cdot p_A(seq), \quad \text{and}$$

$$\sum_{s=0}^{2^k-1} \sum_{seq \in SEQ} p(s<<seq) \cdot p_A(seq) = \sum_{seq \in SEQ} \left[\sum_{s=0}^{2^k-1} p(s<<seq) \right] p_A(seq).$$

Since $\sum_{s=0}^{2^k-1} p(s<<seq) = N(seq)$,

it follows from (3.1.1) and (3.1.2) that $ESC_A(T) = 2^{-k} \cdot \sum_{s=0}^{2^k-1} p_{A,s}(T)$. □

Notice that the proof of Theorem 1 does not use any fact that is unique to circular paths. This means Theorem 1 can be reformulated in a more general form:

Theorem 1':

For any test pattern generation technique, the ESC is equal to the average probability that a state is covered during a testing session. □

In the special case where, for a given test sequence seq, p(seq) = 1, the ESC is the ratio of the number of distinct states in seq over the number of all possible states, while the probability that a state is covered by seq is either 0 or 1.

3.2. APPROXIMATION FORMULA

Based on the fact that the ESC is almost independent of the basic architectural and functional characteristics of the circuit under test, one can estimate the ESC, assuming that the input vectors applied to a circular path section (Fig. 3) are equally likely and independent. This assumption on a circular path will be referred to as the **Equally Likely Independent Input Vectors** (ELIIV) assumption.

Under the ELIIV assumption, if the initial state of the path is random, then the subsequent states of a circular path section are independent and, at any time during a testing session, all states are equally likely. Thus, the ESC depends on the number of flip-flops in A, but not on the particular flip-flop selection. This means that for a set A of k circular path flip-flops $ESC_A(T) = ESC(k,T)$, where ESC(k,T) denotes the expected state coverage for k flip-flops and time T.

From Theorem 1, it follows that ESC(k,T) = p(k,T), where p(k,T) is the probability of covering an arbitrary k-bit vector (state) during T consecutive clock cycles. Since, under the ELIIV assumption, $p(k,T) = 1 - (1 - 2^{-k})^T$ holds, one gets

$$ESC(k,T) = 1 - (1 - 2^{-k})^{T} . \qquad (3.2.1)$$

Now we will find a lower bound on the ESC. Formula (3.2.1) can be transformed as follows :

$$ESC(k,T) = 1 - (1 - 2^{-k})^{T} = 1 - (1 - 2^{-k})^{2^k (T/2^k)} = 1 - [(1 - 2^{-k})^{2^k}]^{(T/2^k)}.$$

Since $(1 - \frac{1}{n})^n$ is a monotonically increasing function of n which tends to e^{-1} as n tends to infinity, one obtains

$$ESC(k,T) \geq 1 - \exp(-\frac{T}{2^k}) . \qquad (F.1)$$

For the convenience of the reader, $\exp(-\frac{T}{2^k})$ in formula (F.1) has been evaluated for several selected values of $T_n = \frac{T}{2^k}$, i.e., the "normalized" time (see Table 1).

$T_n =$	1	2	4	8	16	32
1–ESC ≤	$3.86\ 10^{-1}$	$1.35\ 10^{-1}$	$1.83\ 10^{-2}$	$3.35\ 10^{-4}$	$1.13\ 10^{-7}$	$1.27\ 10^{-14}$

Table 1. Upper bounds on 1–ESC for the ELIIV model.

It must be emphasized that the upper bounds on 1-ESC given in Table 1 have been derived under the assumption that, for a given set of flip-flops, all states on these flip-flops are covered, during a testing session, with the same probability. This follows from the ELIIV assumption. However, it is not valid in general, as illustrated in the following example.

EXAMPLE:

Consider a CSTP section consisting of two adjacent cells. Assume that the section inputs (the input vector in Fig. 3) are both tied to 0, so that the section functions as a 2-bit shift register. For three consecutive clock cycles there are 16 possible sequences of states at the outputs of the 2-bit section of the path. Table 2 lists all these sequences and the number of different states (2-bit vectors) that each sequence covers.

00, 00, 00	1	01, 10, 00	3	10, 00, 00	2	11, 10, 00	3
00, 00, 01	2	01, 10, 01	2	10, 00, 01	3	11, 10, 01	3
00, 01, 10	3	01, 11, 10	3	10, 01, 10	2	11, 11, 10	2
00, 01, 11	3	01, 11, 11	2	10, 01, 11	3	11, 11, 11	1

Table 2. 2-bit shift register sequences for 3 clock cycles and the numbers of states they cover.

Assume that the initial state of the path is random, and that the probability of a 1 at the output of the cell that precedes the section being considered is equal to 0.5. Then, the 16 sequences in Table 2 are equally likely, and one can easily find that $ESC(2,3) = 2^{-2} \cdot \frac{38}{16} = \frac{19}{32}$, $p_0(3) = \frac{8}{16}$, $p_1(3) = \frac{11}{16}$, $p_2(3) = \frac{11}{16}$, and $p_3(3) = \frac{8}{16}$, where $p_i(3)$ denotes the probability of covering state i, i= 0, 1, 2, 3, when T = 3. This shows that, although all states are equally likely for each clock cycle, the probability of covering different states may differ, e.g., $p_0(3) \neq p_1(3)$. The average probability is, however, equal to the ESC as stated by Theorem 1.□

3.3. SIMULATION STUDY USING A PROBABILISTIC MODEL

Formula (F.1) has been developed under the ELIIV assumption. One may ask if this lower bound on the ESC holds for circuits for which the ELIIV assumption does not hold. As a precise analysis of this problem seems to be extremely difficult, we have conducted simulation studies. Surprisingly, it turns out that (F.1) provides a very accurate estimate for the ESC (not just a lower bound).

3.3.1. PREVIOUS MODEL

The first simulation results on the CSTP test generation effectiveness were reported in [KRPI87, KRPI89], where the dependency of the ESC on a number of factors was investigated. These results were based on a simple model that considered the outputs of a combinational block as a set of one-bit independent random variables. Each of these variables was assumed to take value 1 with a fixed probability. Unfortunately, in real circuits the output variables of a combinational block may be strongly correlated. An example of a simple circuit with strongly correlated output variables was given in [KRPI89]. That particular example indicated that the ESC was not affected by the correlation between output variables. Nevertheless, to verify this observation, we decided to develop a circuit model that allows such correlation.

3.3.2. NEW MODEL

Consider a section of a CSTP that consists of a number of adjacent cells. During testing, the section inputs (see Fig. 3) are stimulated by the outputs of combinational block CL1, while the output vectors produced by the section constitute test patterns for combinational block CL2. A given combinational block, e.g., CL1, may not output all possible binary vectors, i.e., a fault-free circuit with k outputs has a set of possible responses that, in general, constitute a subset of all binary vectors of length k. During testing, the inputs to CL1 (not shown in Fig.3) are stimulated by vectors that are characterized by a probability distribution. This probability distribution and the function of CL1 determine the probability distribution of the CL1's output vectors. Thus, in our new model used for the simulation sstudy presented below, the section input vectors are characterized by a probability distribution. The validity of this model is verified in Section 3.4, where some results presented below are compared with the results obtained from sample circuit simulations.

3.3.3. RESULTS

For each simulation experiment, the length of the section, k, and the number of different vectors that are applied to the inputs to the circular path section, v ($v < 2^k$), are specified. In a single simulation run, a set of v vectors that can appear at the CSTP section inputs, along with their probabilities are randomly generated. Based on the probability distribution of the input vectors, the matrix of state transition probabilities is calculated, where a state is defined by the contents of the k flip-flops constituting the CSTP section. This matrix, in turn, is used for computing the probability that a particular state is not covered within T clock cycles. Finally, using Theorem 1, the ESC is calculated as the average of the probabilities of covering all possible states.

A number of simulations were run with different sets of v input vectors and different distributions of their probabilities. The mean ESC and its standard deviation were then computed. Table 3 and 4 present the results for a CSTP section consisting of 6 cells (k=6). Each entry in Table 3 contains the average of 1–ESC for 25 runs for the same values of T and v, while the corresponding entry of Table 4 shows the standard deviation. Both tables use normalized time T_n, i.e., $\frac{T}{2^k}$.

v \ T_n	1	2	4	8	16	32
64	$3.65\ 10^{-1}$	$1.33\ 10^{-1}$	$1.77\ 10^{-2}$	$3.15\ 10^{-4}$	$9.96\ 10^{-8}$	$1.00\ 10^{-14}$
48	$3.65\ 10^{-1}$	$1.33\ 10^{-1}$	$1.77\ 10^{-2}$	$3.16\ 10^{-4}$	$1.00\ 10^{-7}$	$1.04\ 10^{-14}$
32	$3.65\ 10^{-1}$	$1.33\ 10^{-1}$	$1.78\ 10^{-2}$	$3.17\ 10^{-4}$	$1.02\ 10^{-7}$	$1.13\ 10^{-14}$
24	$3.65\ 10^{-1}$	$1.33\ 10^{-1}$	$1.78\ 10^{-2}$	$3.18\ 10^{-4}$	$1.04\ 10^{-7}$	$1.24\ 10^{-14}$
16	$3.65\ 10^{-1}$	$1.33\ 10^{-1}$	$1.78\ 10^{-2}$	$3.20\ 10^{-4}$	$1.08\ 10^{-7}$	$1.55\ 10^{-14}$

Table 3. Mean of 1–ESC, k=6.

v \ T_n	1	2	4	8	16	32
64	$1.88\ 10^{-6}$	$2.50\ 10^{-8}$	$7.32\ 10^{-7}$	$7.90\ 10^{-8}$	$1.18\ 10^{-10}$	$5.13\ 10^{-17}$
48	$4.05\ 10^{-6}$	$6.89\ 10^{-8}$	$1.58\ 10^{-6}$	$1.72\ 10^{-7}$	$2.62\ 10^{-10}$	$1.20\ 10^{-16}$
32	$8.27\ 10^{-6}$	$2.42\ 10^{-7}$	$3.21\ 10^{-6}$	$3.63\ 10^{-7}$	$5.88\ 10^{-10}$	$3.14\ 10^{-16}$
24	$1.51\ 10^{-5}$	$6.44\ 10^{-7}$	$5.88\ 10^{-6}$	$6.96\ 10^{-7}$	$1.23\ 10^{-9}$	$8.03\ 10^{-16}$
16	$2.60\ 10^{-5}$	$1.77\ 10^{-6}$	$1.01\ 10^{-5}$	$1.30\ 10^{-6}$	$2.78\ 10^{-9}$	$2.92\ 10^{-15}$

Table 4. Standard deviation, k=6.

Comparing these results with the values in Table 1, one can easily notice that formula (F.1) gives very accurate approximations to the values presented in Table 3. The extremely small values of standard deviation assure that the ESC is practically independent of the circuit characteristics. Based on many experiments, we make the following general observation:

Observation:

Formula (F.1) can be used to estimate the ESC of any CSTP section whose inputs come from the outputs of a block (or blocks) under test. □

The values of 1–ESC presented in Table 3 indicate that, for a fixed T_n, the ESC slightly decreases with decreasing v. In fact, the lowest ESC was observed when v = 1, regardless of the CSTP section length. Table 5 shows 1–ESC for several k, when the CSTP section inputs were fixed to constant values, i.e., v = 1.

A graphical representation for the degradation of ESC with decreasing values of v for k = 6 is given in Fig. 4, where $\ln(1-ESC(T/2^k))$ is plotted against normalized time.

k \ T_n	1	2	4	8	16	32
4	$3.43\ 10^{-1}$	$1.24\ 10^{-1}$	$2.06\ 10^{-2}$	$1.21\ 10^{-3}$	$5.70\ 10^{-6}$	$7.65\ 10^{-10}$
6	$3.61\ 10^{-1}$	$1.32\ 10^{-1}$	$1.92\ 10^{-2}$	$7.30\ 10^{-4}$	$6.84\ 10^{-6}$	$1.45\ 10^{-9}$
8	$3.66\ 10^{-1}$	$1.34\ 10^{-1}$	$1.86\ 10^{-2}$	$4.55\ 10^{-4}$	$2.42\ 10^{-6}$	$6.68\ 10^{-10}$
12	$3.68\ 10^{-1}$	$1.35\ 10^{-1}$	$1.83\ 10^{-2}$	$3.44\ 10^{-4}$	$2.77\ 10^{-7}$	$5.36\ 10^{-11}$
16	$3.68\ 10^{-1}$	$1.35\ 10^{-1}$	$1.83\ 10^{-2}$	$3.36\ 10^{-4}$	$1.30\ 10^{-7}$	$3.44\ 10^{-12}$

Table 5. 1 – ESC, for the case v=1.

It is seen that 1–ESC for v = 1 is larger than 1–ESC for larger values of v. Of course, v = 1 means that the inputs of the CSTP section are not connected to the outputs of any block under test. It turns out, however, that for larger values of k, the degradation is insignificant from a practical point of view. For example, for k = 16, the differences between the exact values of ESC and those predicted by (F.1) are negligible for $T_n \leq 16$.

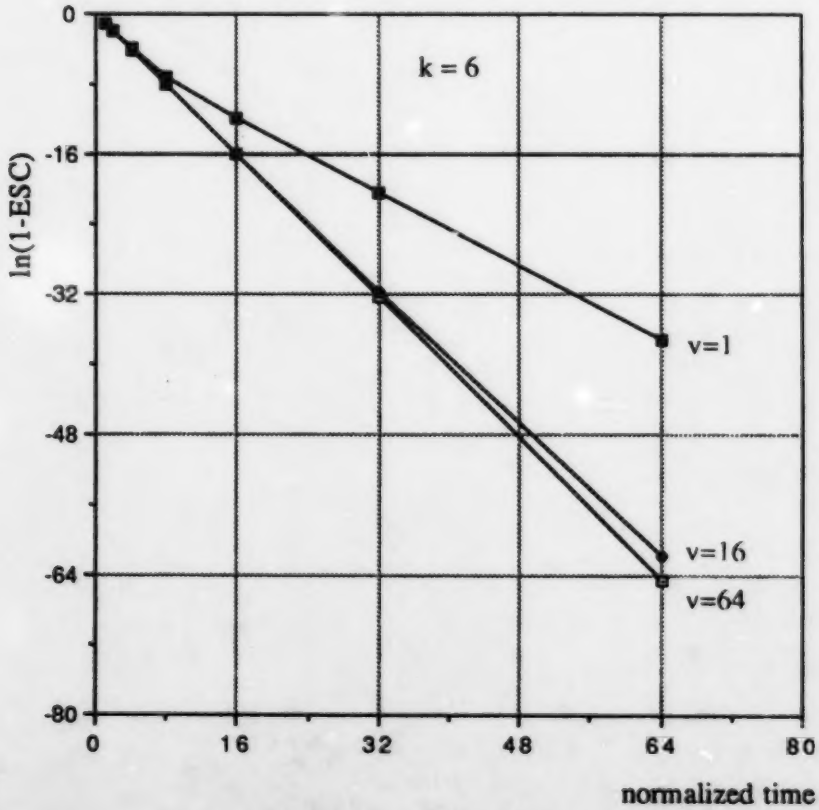

Fig. 4. Degradation of ESC for k = 6.

It is worth pointing out that the results presented in Table 5 were obtained using a slightly different simulation method than that used to obtain the results presented in Table 3. The previous algorithm, because of its complexity, allowed us to compute the ESC for CSTP sections of length up to 7. For v = 1 we were able to compute the ESC for CSTP sections of length up to 16. The modification to the simulation method is based on the observation that, for v = 1, the ESC does not depend on the vector applied to the CSTP inputs. In fact, it is simple to show that if v = 1, then for any two input vectors applied to the CSTP section the state transition graphs are isomorphic. For a section input vector consisting of all zeros, the CSTP section obviously constitutes a shift register. When any other vector is applied to the section inputs, the state bit pattern is modified, but this modification is a permutation.

The state transition matrix that we used in Section 3.3.2 had 2^k rows and 2^k columns, since a CSTP section consisting of k flip-flops has 2^k different states. Here, we partition the states into k+1 equivalence classes and consider each class as a single state of a finite automaton. Fig. 5 illustrates such an automaton for a 3-bit register (k = 3) and test pattern <110>. This type of finite automaton plays a key role in the proof of the formula presented in the appendix, where we consider it more formally. Below, we present an introduction to this model.

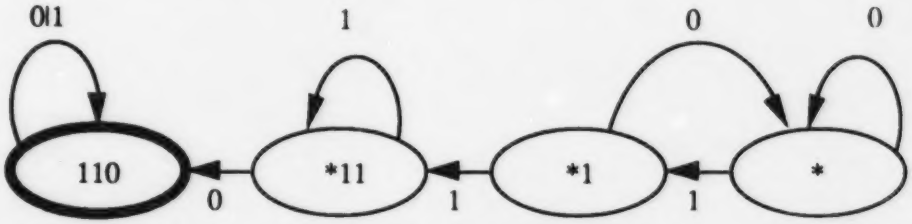

Fig. 5. Finite automaton for state <110>.

According to our convention, the contents of a shift register is always shifted to the left. The leftmost state (accepting state) of the automaton in Fig. 5 represents the situation where vector <110> has already been covered, i.e., it has already appeared in the shift register at least once. The rest of the states represent the situation where vector <110> has not been covered. The second state from the left, for example, represents the equivalence class consisting of two states <011> and <111>. The last two digits, namely <11>, constitute a prefix of vector <110>, and if the next bit shifted in is a 0, a transition to the accepting state occurs. However, if the next bit is a 1, then the two-digit suffix of the input sequence, namely <11>, constitutes a prefix of vector

<110>, and, therefore, a transition to state *11 occurs. The other states and transitions can be constructed in a similar way. Under the assumption that 0 and 1 are shifted in with equal probability, the probability of covering state <110> can be easily computed as the probability of being in the accepting state after T–1 transitions. The initial probabilities of the states of the automaton in Fig. 5 are 2^{-3}, 2^{-2}, 2^{-2}, $3 \cdot 2^{-3}$, respectively. They are determined using the number of states in each equivalence class. For instance, the accepting state, representing a one-member equivalence class, has the initial probability equal to 2^{-3}. The initial probability distribution can also be computed assuming that at time T = –k+1, the probability of the rightmost state (state *) is 1.

3.4. SIMULATION USING SAMPLE CIRCUITS

To verify the applicability of our formula and results already presented, we checked the CSTP state sequences for a class of circuits consisting of three functional blocks shown in Fig. 6. Each functional block has six inputs and six outputs (represented by thin arrows).

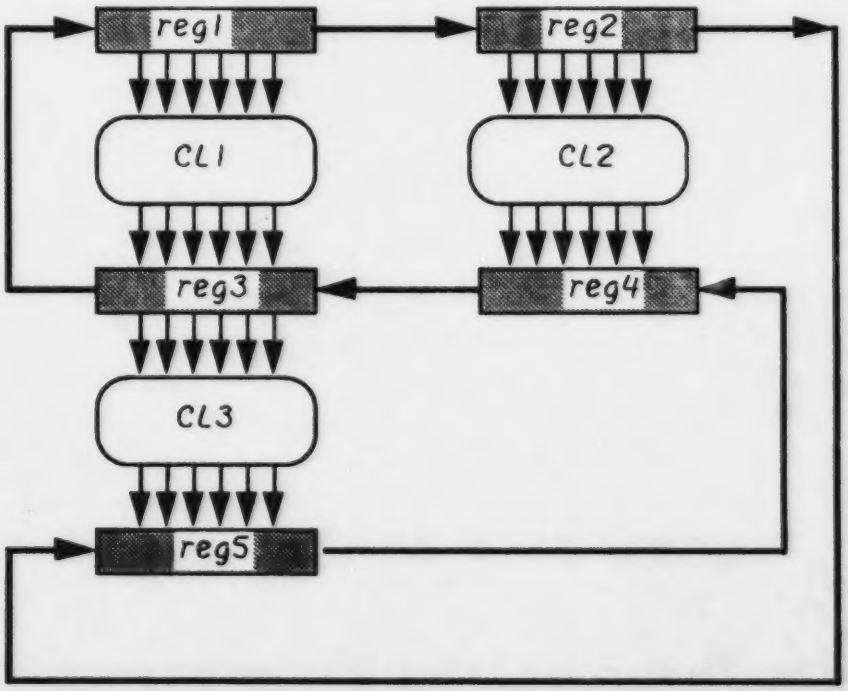

Fig. 6. Sample circuit.

For each simulation run, the input/output mapping was randomly selected for each combinational block, and the exact state coverage as a function of time was checked at a k-bit CSTP section. The function of a combinational block, when generated, was assumed to be implemented as a read-only memory, and each bit of the memory was set to 1 with a predefined probability, p, that characterizes the circuit. Tables 6, 7 and 8 summarize some of the results. Each entry is the mean of 10000 simulation runs. Table 6 shows the results for an 8-bit CSTP section whose inputs were stimulated by outputs of a CUT. Tables 7 and 8 present the results for a CSTP section with v = 1 for two different values of k (8 and 4, respectively). ASC stands for average state coverage.

p T_n	1	2	4	8
0.1	$3.67\ 10^{-1}$	$1.35\ 10^{-1}$	$1.83\ 10^{-2}$	$3.45\ 10^{-4}$
0.3	$3.67\ 10^{-1}$	$1.35\ 10^{-1}$	$1.82\ 10^{-2}$	$3.34\ 10^{-4}$
0.5	$3.67\ 10^{-1}$	$1.35\ 10^{-1}$	$1.81\ 10^{-2}$	$3.09\ 10^{-4}$
0.7	$3.67\ 10^{-1}$	$1.35\ 10^{-1}$	$1.81\ 10^{-2}$	$3.22\ 10^{-4}$
0.9	$3.67\ 10^{-1}$	$1.35\ 10^{-1}$	$1.83\ 10^{-2}$	$3.54\ 10^{-4}$
mean	$3.67\ 10^{-1}$	$3.67\ 10^{-1}$	$1.82\ 10^{-2}$	$3.33\ 10^{-4}$

Table 6. Mean of 1–ASC, stimulated, for k=8.

p T_n	1	2	4	8
0.1	$3.66\ 10^{-1}$	$1.35\ 10^{-1}$	$1.86\ 10^{-2}$	$4.66\ 10^{-4}$
0.3	$3.66\ 10^{-1}$	$1.35\ 10^{-1}$	$1.86\ 10^{-2}$	$4.75\ 10^{-4}$
0.5	$3.66\ 10^{-1}$	$1.34\ 10^{-1}$	$1.86\ 10^{-2}$	$4.51\ 10^{-4}$
0.7	$3.66\ 10^{-1}$	$1.34\ 10^{-1}$	$1.86\ 10^{-2}$	$4.82\ 10^{-4}$
0.9	$3.66\ 10^{-1}$	$1.35\ 10^{-1}$	$1.86\ 10^{-2}$	$4.67\ 10^{-4}$
mean	$3.66\ 10^{-1}$	$1.35\ 10^{-1}$	$1.86\ 10^{-2}$	$4.68\ 10^{-4}$

Table 7. Mean of 1–ASC for v=1 and k=8.

p T_n	1	2	4	8
0.1	$3.42\ 10^{-1}$	$1.24\ 10^{-1}$	$2.02\ 10^{-2}$	$1.13\ 10^{-3}$
0.3	$3.43\ 10^{-1}$	$1.24\ 10^{-1}$	$2.17\ 10^{-2}$	$1.31\ 10^{-3}$
0.5	$3.43\ 10^{-1}$	$1.23\ 10^{-1}$	$2.03\ 10^{-2}$	$1.16\ 10^{-3}$
0.7	$3.43\ 10^{-1}$	$1.24\ 10^{-1}$	$2.06\ 10^{-2}$	$1.16\ 10^{-3}$
0.9	$3.45\ 10^{-1}$	$1.25\ 10^{-1}$	$2.09\ 10^{-2}$	$1.25\ 10^{-3}$
mean	$3.43\ 10^{-1}$	$1.24\ 10^{-1}$	$2.07\ 10^{-2}$	$1.20\ 10^{-3}$

Table 8. Mean of 1–ASC for v=1 and k=4.

It should be pointed out that the results presented in this section are practically identical to those obtained by simulation using our circuit probabilistic model (see Section 3.3.3). The similarity may be surprising, especially if we consider the fact that the model that we used in Section 3.3.3 makes a simplistic assumption about the feedback that exists in circular paths. The results shown in Tables 7 and 8 are in many cases indistinguishable from the results presented in Table 5 (rows k=8 and k=4). Similarly, the results shown in Table 6 do not depend on the probability p characterizing the circuit, and confirm the applicability of formula (F.1) for estimating the ESC for a CSTP section whose inputs are stimulated by outputs of a block under test (see Fig. 7). Thus, the formula (F.1) can be used in practice to estimate the ESC.

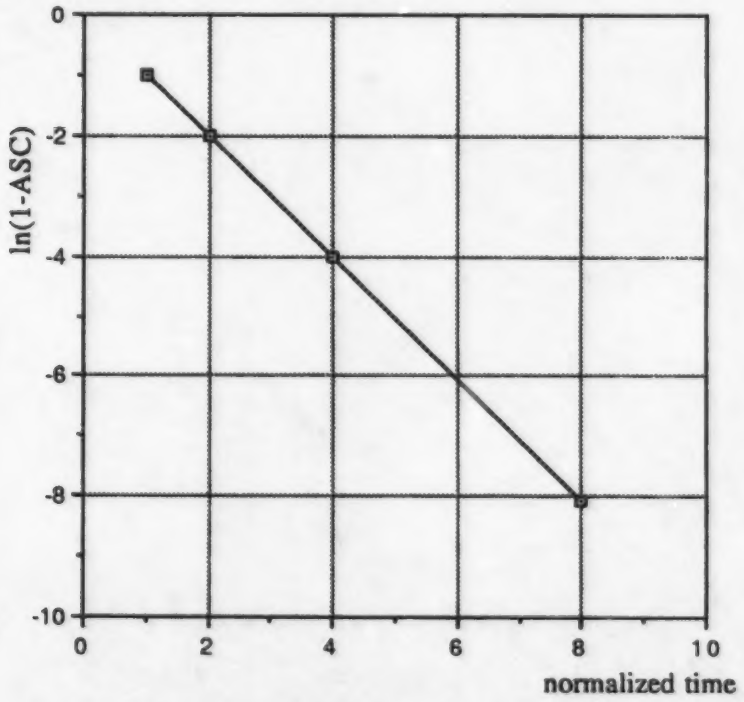

Fig. 7. Statistical data for k=8 practically identical to results predicted by formula (F.1).

4. THE LEAST PROBABLE STATE

As shown in the previous section, in the case where the inputs of a CSTP section are connected to the outputs of blocks under test, formula (F.1) can be used to estimate the ESC. Unfortunately, a similar statement is not, in general, true for v = 1.

In this case, as commented in Section 3.3.3, the applicability of formula (F.1) is limited to larger k's and to relatively small values of T_n. Thus, we do not have any general formula to estimate the ESC if the inputs of a CSTP section are not connected to the outputs of blocks under test. Since this situation is commonly encountered in practice (e.g. circuit primary input flip-flops are included in CSTP), it is important to find a method that makes such an estimation possible. Below, for the case v=1, we present a formula for the probability of covering the least probable state. By Theorem 1, this probability constitutes a lower bound on the ESC.

As shown in Theorem 2 in the appendix, if a CSTP section is equivalent to a shift register, then, regardless of the length of the register and testing time, state 0 is the least probable. The probability of covering this state can easily be computed using the finite automaton model introduced in the previous section.

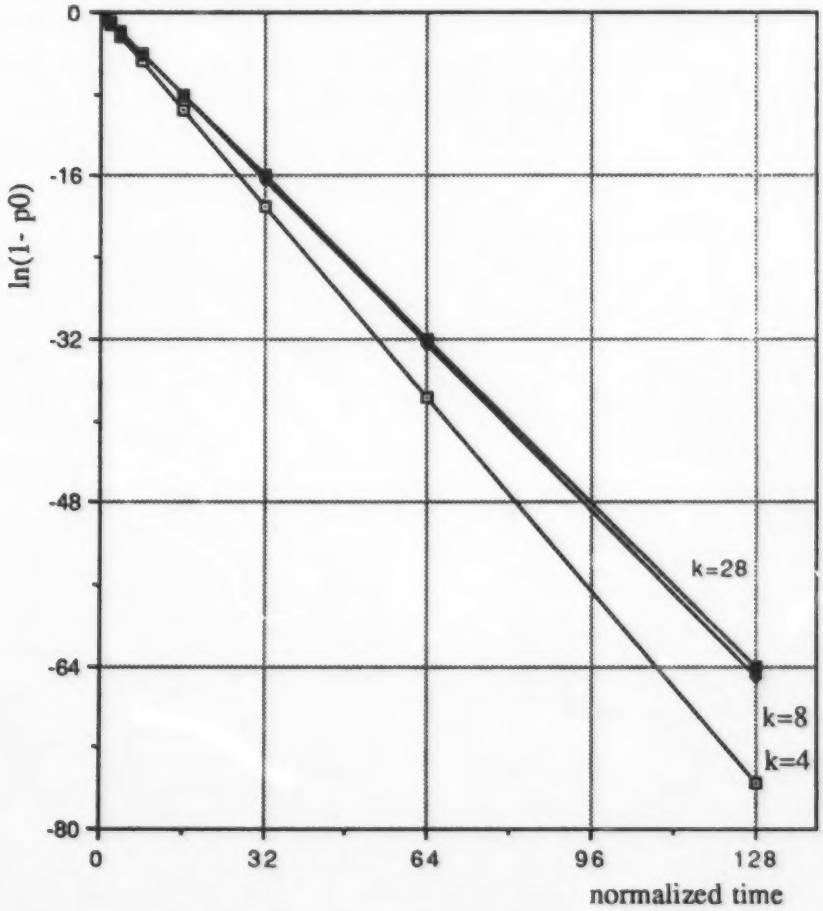

Fig. 8. Probability that state 0 is covered.

Some simulation results illustrated in Fig. 8 indicate that there may exist a nontrivial lower bound on the probability of covering the least probable state. In fact, one can conjecture that

$$1 - \exp(-T/2^{k+1}) \qquad (F.2)$$

would be such a bound. A proof that this conjecture holds is presented in the appendix.

5. PROBABILITY OF ALIASING

The probability of aliasing for LFSR-based test response compaction has been studied by many researchers, e.g.,[WDGS88, DOFR89, IVAG89]. From the CSTP point of view, the most interesting results are those concerning multiple input compaction [WIDA89, DOFER89, MIZH89], since a circular path, when feedback through the blocks under test is ignored, can be considered as a LFSR with a characteristic polynomial $x^P \oplus 1$, where P is the path length. Unfortunately, the existence of feedback makes these results invalid for CSTP aliasing estimation. The reason for this is that the error models used for aliasing analysis do not take into account the fact that the probability of an error changes in time or the fact that the error streams are correlated. The first error, when propagated to the inputs of any block under test, affects error probabilities at this block's outputs. This, in turn, affects other blocks and usually changes the circular path state in a random-like way. This phenomenon, unfortunate from an analysis point of view, reduces aliasing in a CSTP.

In [KRPI89], it was shown that the probability that a circular path flip-flop takes value 1 at any time instant during a testing session can be assumed to be 0.5. Our intuition says that, practically, there will be no correlation between the signature produced by a faulty circuit and the fault-free signature. This implies that the probability of aliasing may be approximated by expression 2^{-S}, where S is the length of the signature. A formal proof of this conjecture seems to be extremely difficult, because of the nonlinear feedback through combinational logic, and still remains an open problem. However, to examine whether our intuition is correct, extensive simulation studies were performed. Before we describe our simulation results, for the convenience of the reader, we list some approximate values of 2^{-S} (Tab. 9).

S =	4	6	8	10	12
$2^{-S} \approx$	$6.25\ 10^{-2}$	$1.56\ 10^{-2}$	$3.91\ 10^{-3}$	$9.77\ 10^{-4}$	$2.44\ 10^{-4}$

Table 9. Conjectured values of the probability of aliasing.

Each circuit we simulated consists of two pipelines, each consisting of two combinational blocks and three registers, as shown in Fig. 9. Each combinational block has five inputs and five outputs.

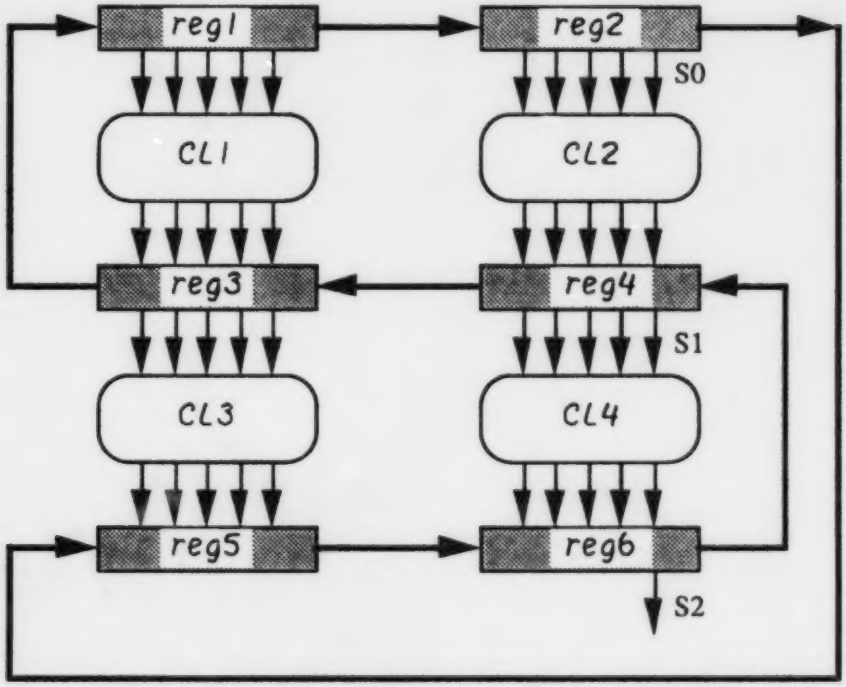

Fig. 9. Two pipelines tested by a circular path.

For each simulation run, the functions of the combinational blocks were randomly selected, then a stuck-at fault was injected at a specific location of the circuit. Each combinational block was assumed to be implemented as a read-only memory and, when generating the block function, each bit of the memory was set to 1 with a predefined probability, p. Experiments were conducted for three categories of faults:

C1 stuck-at fault at a CSTP flip-flop output,

C2 stuck-at fault at a combinational block output, and

C3 stuck-at fault of exactly one bit of read only memory.

For category C3, a stuck-at fault was injected by changing a 0 to a 1, or 1 to 0, depending on the fault type. The simulator checked whether the fault was sensitized, i.e., whether the circular path generated a test for the fault, and if the signature

produced by the faulty circuit was equal to the fault-free value. The test running time was set to $8 * 2^5$ clock cycles (i.e.,T_n=8). The signatures were collected at three different locations (called signature points).

For a given fault (type and location), signature point, value of p, and testing session length, the simulator computed the fraction of 409600 simulation runs that resulted in fault free signatures,[4] FFS, and the fraction of faults that were not sensitized, FNS. Finally, it computed the value of expression $\frac{FFS-FNS}{1-FNS}$. We call this expression **aliasing ratio**; it represents the probability of aliasing as defined in [DAV90]. Table 10 presents the results for a stuck-at-1 fault injected at the output of the leftmost flip-flop of register *reg1*; signatures were collected at point S0 for several values of p and S. The mean values of these results, also presented in Table 10, are plotted in Fig. 10. As seen in the figure they are practically identical to the results predicted by formula 2^{-S}.

Space limitation prevents us from listing here all the results of the simulations that we ran. It should, however, be stated that the results presented in Table 10 are typical of faults of categories C1 and C2, and there is no significant difference among the results for several signature points.

p \ S	4	6	8	10	12
0.1	$6.21\ 10^{-2}$	$1.59\ 10^{-2}$	$3.96\ 10^{-3}$	$1.00\ 10^{-3}$	$2.22\ 10^{-4}$
0.3	$6.23\ 10^{-2}$	$1.55\ 10^{-2}$	$3.84\ 10^{-3}$	$9.01\ 10^{-4}$	$2.27\ 10^{-4}$
0.5	$6.31\ 10^{-2}$	$1.55\ 10^{-2}$	$3.89\ 10^{-3}$	$9.45\ 10^{-4}$	$2.73\ 10^{-4}$
0.7	$6.23\ 10^{-2}$	$1.54\ 10^{-2}$	$3.88\ 10^{-3}$	$1.01\ 10^{-3}$	$2.49\ 10^{-4}$
0.9	$6.32\ 10^{-2}$	$1.57\ 10^{-2}$	$3.94\ 10^{-3}$	$9.98\ 10^{-4}$	$2.51\ 10^{-4}$
mean	$6.26\ 10^{-2}$	$1.56\ 10^{-2}$	$3.90\ 10^{-3}$	$9.70\ 10^{-4}$	$2.44\ 10^{-4}$

Table 10. Stuck-at-1 at the output of circular path flip-flop: signatures collected at point S0.

[4]This fraction measures the quality of testing since it reflects both test generation and response compaction.

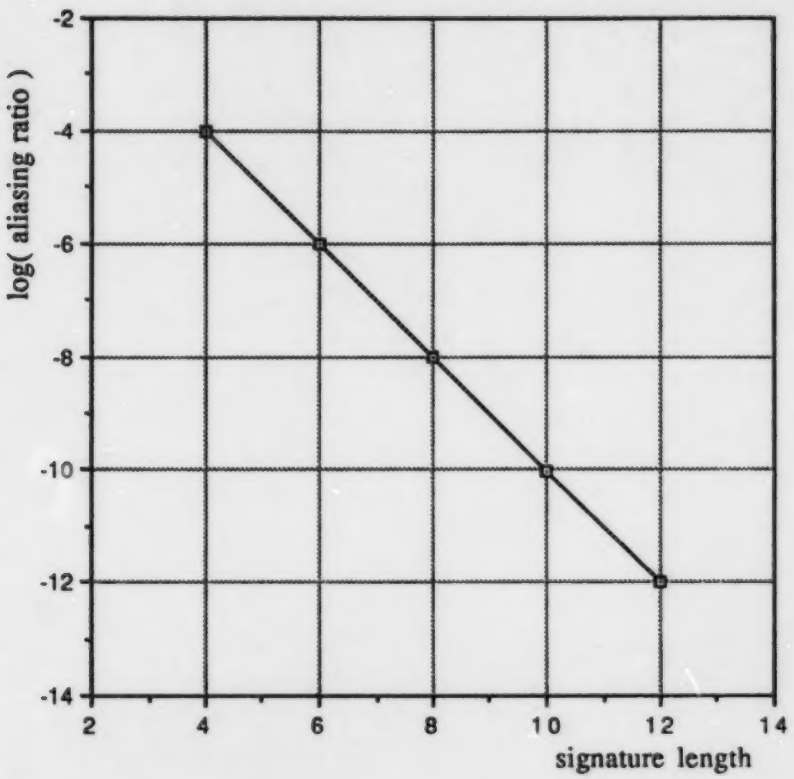

Fig. 10. Aliasing ratio for faults of category C1.

The typical results for faults of category C3 are slightly different from those for C1 and C2. Table 11 shows the results for faults that changed one bit in the memory implementing the function of block CL1. These results are typical of fault category C3.

p \ S	4	6	8	10	12
0.1	$6.27\ 10^{-2}$	$1.54\ 10^{-2}$	$3.95\ 10^{-3}$	$1.03\ 10^{-3}$	$3.08\ 10^{-4}$
0.3	$6.23\ 10^{-2}$	$1.57\ 10^{-2}$	$3.91\ 10^{-3}$	$9.99\ 10^{-4}$	$3.03\ 10^{-4}$
0.5	$6.29\ 10^{-2}$	$1.55\ 10^{-2}$	$3.75\ 10^{-3}$	$1.02\ 10^{-3}$	$2.56\ 10^{-4}$
0.7	$6.30\ 10^{-2}$	$1.57\ 10^{-2}$	$3.93\ 10^{-3}$	$9.94\ 10^{-4}$	$2.88\ 10^{-4}$
0.9	$6.24\ 10^{-2}$	$1.57\ 10^{-2}$	$3.79\ 10^{-3}$	$1.00\ 10^{-3}$	$3.18\ 10^{-4}$
mean	$6.26\ 10^{-2}$	$1.56\ 10^{-2}$	$3.87\ 10^{-3}$	$1.00\ 10^{-3}$	$2.95\ 10^{-4}$

Table 11. Signatures collected at point S1 for stuck-at-1 faults injected inside block CL1.

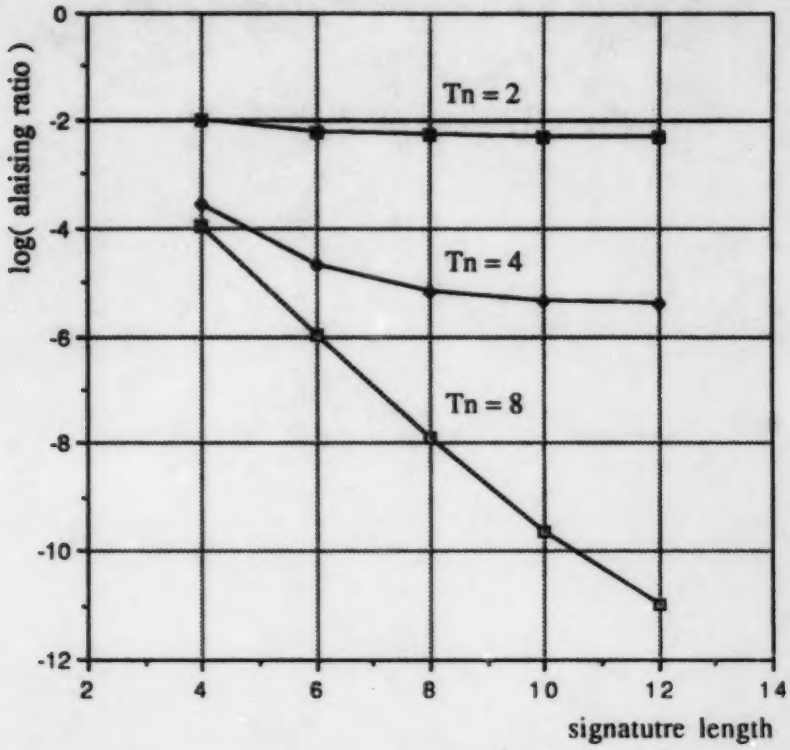

Fig. 11. Aliasing ratio versus signature length:
signature point S0, faults of category C3 .

Fig. 11 illustrates aliasing ratios for different T (T_n = 2, 4, and 8) for signatures collected at point S0. Fig. 12 shows the difference between the aliasing ratios at signature points S0, S1 and S2 for a testing session of length $T = 2 * 2^5$ clock cycles ($T_n = 2$). One can notice that for longer signatures, the aliasing ratio for faults of category C3 is slightly larger than that for faults of category C1 or C2 (compare the curve of Fig. 10 with curve T_n=8 in Fig. 11). The reason for this is the fact that relatively many faults were sensitized for the first time near the end of a testing session and there was not enough time for the error to affect the signature. For the sample circuit shown in Fig. 9, if the leftmost variable of CL3 is affected by a fault of category C3, then an error produced by the fault needs 9, 10 and 29 clock cycles to affect signatures collected at point S2, S1 and S0, respectively. One may expect that this effect is more significant for short testing sessions. Figs. 11 and 12 illustrate this phenomenon. The data were computed for faults injected in CL3 which affected each output with the same probability.

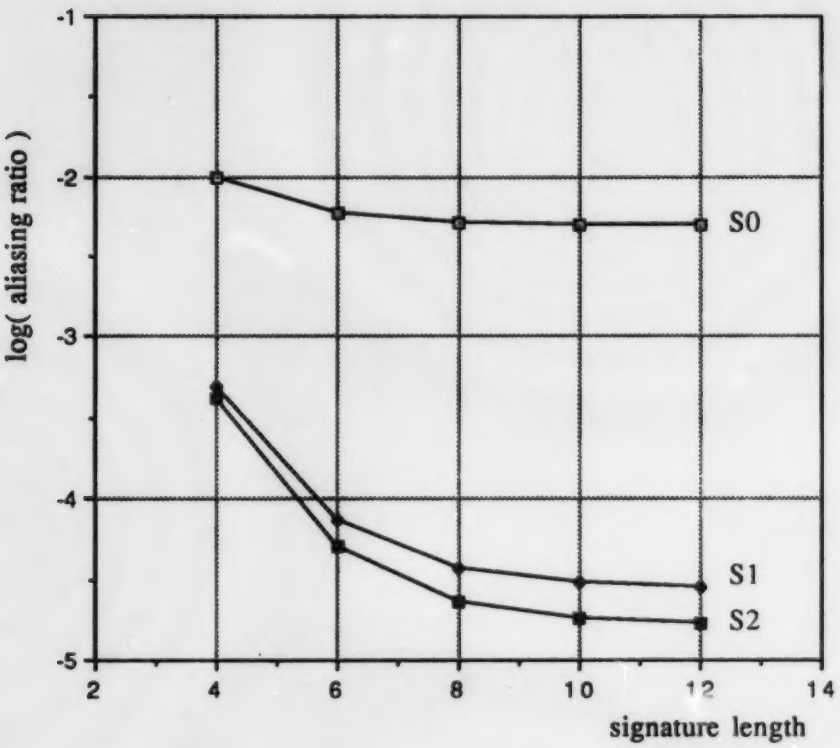

Fig. 12. Aliasing ratio at different signature points.

Now we will estimate the factor in the aliasing probability that is due to the fact that it takes some time for the first error to affect the signature. Assume that L+1 is the number of clock cycles that an error caused by fault f needs to affect the signature,[5] and let q_f be the probability that fault f is sensitized at a given clock cycle. The probability that fault f is sensitized for the first time within the last L clock cycles of the testing session equals $(1-q_f)^{T-L} - (1-q_f)^T$. Thus the probability of aliasing caused by this is $\frac{(1-q_f)^{T-L} - (1-q_f)^T}{1-(1-q_f)^T}$, where the denominator expresses the probability that the fault was sensitized during a testing session. The event represented by this probability is mutually exclusive from the event that f is sensitized within the initial T–L clock cycles. The probability of the latter is given by

---

[5] L cannot be longer than P (the total number of flip-flops in the circular path).

$1-(1-q_f)^{T-L}$. Therefore, based on our conjecture on aliasing probability stated in the beginning of this section, the probability of aliasing for fault f can be estimated by $\frac{(1-q_f)^{T-L}-(1-q_f)^{T}}{1-(1-q_f)^{T}}+\frac{1-(1-q_f)^{T-L}}{1-(1-q_f)^{T}}\cdot 2^{-S}$, which reduces to

$$1-\frac{1-(1-q_f)^{T-L}}{1-(1-q_f)^{T}}\cdot(1-2^{-S}). \qquad \text{(F.3)}$$

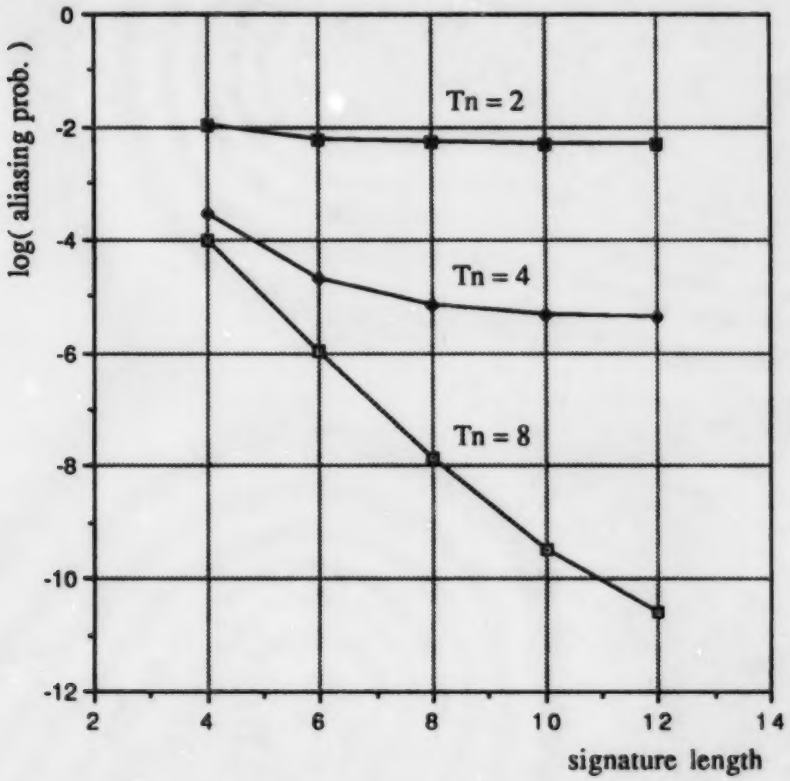

Fig. 13. Aliasing probability versus signature length: signature point S0, faults of category C3.

Let us compare the results illustrated in Figs. 11 and 12 with the values obtained from formula (F.3). Since, for any fault of category C3, there is exactly one test that sensitizes the fault, we have $q_f = 2^{-5}$ for a 5 input combinational circuit. As mentioned before, the simulator injected faults that affected any of CL3's outputs with equal probability. Thus, to calculate the aliasing probability at signature point S0, for example, one has to calculate it for L equal to 29, 28, 27, 26, and 25. The desired probability is equal to the average of these five. Figs. 13 and 14 show that there is no

significant difference between the simulation results and those obtained using formula (F.3). Table 12 presents the simulation results versus those expected by (F.3) for $T_n = 2$, signature point S1, and several signature lengths.

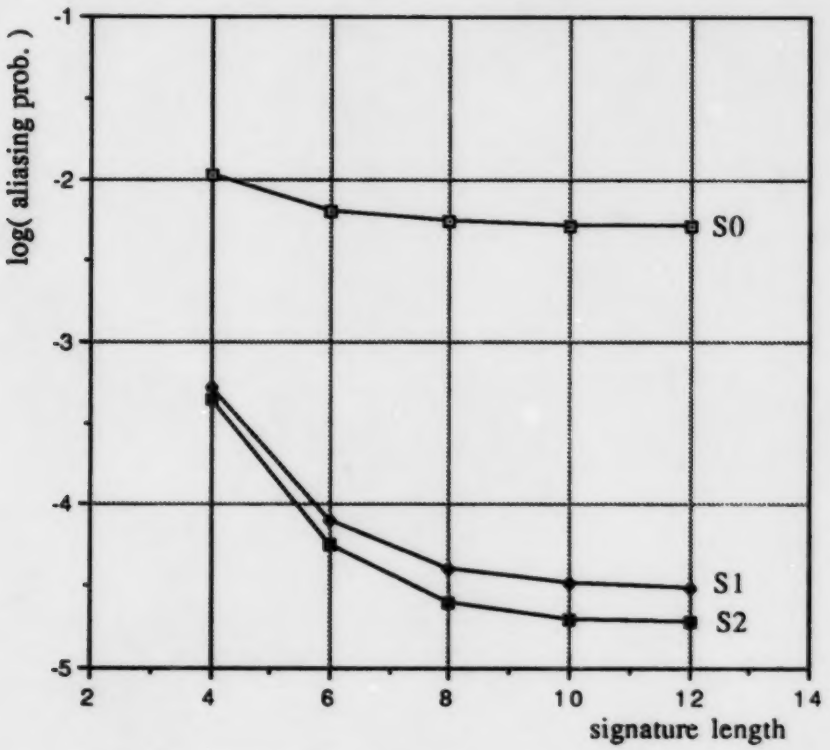

Fig. 14. Aliasing probability at different signature points: faults of category C3.

S =	4	6	8	10	12
simulated	$1.01\ 10^{-1}$	$5.71\ 10^{-2}$	$4.63\ 10^{-2}$	$4.35\ 10^{-2}$	$4.28\ 10^{-2}$
by (F.3)	$1.03\ 10^{-1}$	$5.86\ 10^{-2}$	$4.74\ 10^{-2}$	$4.46\ 10^{-2}$	$4.39\ 10^{-2}$

Table 12. Aliasing ratio versus aliasing probability.

Formula (F.3) represents a decreasing function with respect to the probability q_f (for $q_f \neq 0$ and T>L). Therefore, the aliasing probability is the largest for the hardest-to-detect faults. (F.3) is increasing with L, thus, from the aliasing point of view, the circular path should be designed in a way that minimizes L. Since an error propagates not only along the circular path but also through combinational blocks, to minimize L means, for example, that the input and output registers of a combinational block should not be placed in the circular path next to each other. Fortunately, however, the effect of

L in function (F.3) decreases with T, and in practice, it can be approximated by 2^{-S}. To see this more clearly, consider the following example:

EXAMPLE:

Consider a circuit with a circular path consisting of 1024 flip-flops. Assume that the probability of sensitizing the hardest-to-detect fault, f, is 2^{-16}, and the test running time, T, is $16 * 2^{16}$ clock cycles (a fraction of a second for a typical VLSI chip). The part of probability of aliasing that is independent of signature length (see the formula just above (F.3)) does not exceed 10^{-9} (a very pessimistic evaluation based on the assumption that L = 1024) and is at least two orders of magnitude smaller than the probability that f is not sensitized. Note also that for the majority of faults, it will be orders of magnitude smaller. Similarly, it is seen that the coefficient of $(1 - 2^{-S})$ in (F.3), i.e., $\frac{1-(1-q_f)^{T-L}}{1-(1-q_f)^{T}}$ can be considered as one, so that (F.3) can be replaced by 2^{-S} with little loss in accuracy.□

To verify our conjecture on aliasing probability (2^{-S}) more thoroughly, we have checked aliasing for several benchmark circuits tested using CSTP. Every input and output of each circuit was assumed to have a corresponding flip-flop. All input, output and internal flip-flops (if any) were included in the circular path. The order of flip-flops in the path was assumed to be the same as in the original text file describing the circuit [BRFU85]. All signatures were collected at the same signature point. The initial state of the path was selected by a random number generator. Our simulator, for a given testing time and signature length, determined the fault-free signature, and then examined the signature for each single stuck-at fault in the combinational logic. Fig. 15 illustrates the aliasing ratios for benchmark circuit c880 [BRFU85] for several lengths of testing session and signature lengths. The number of circular path flip-flops in this circuit is 86 (60 inputs and 26 outputs). Our simulator reported that circuit c880 has one output which depends on 45 inputs. There are 2224 possible single stuck-at faults. It is seen in Fig. 15 that, when time T is small, the effect of L is significant. However, as T increases, the aliasing ratios approach 2^{-S}.

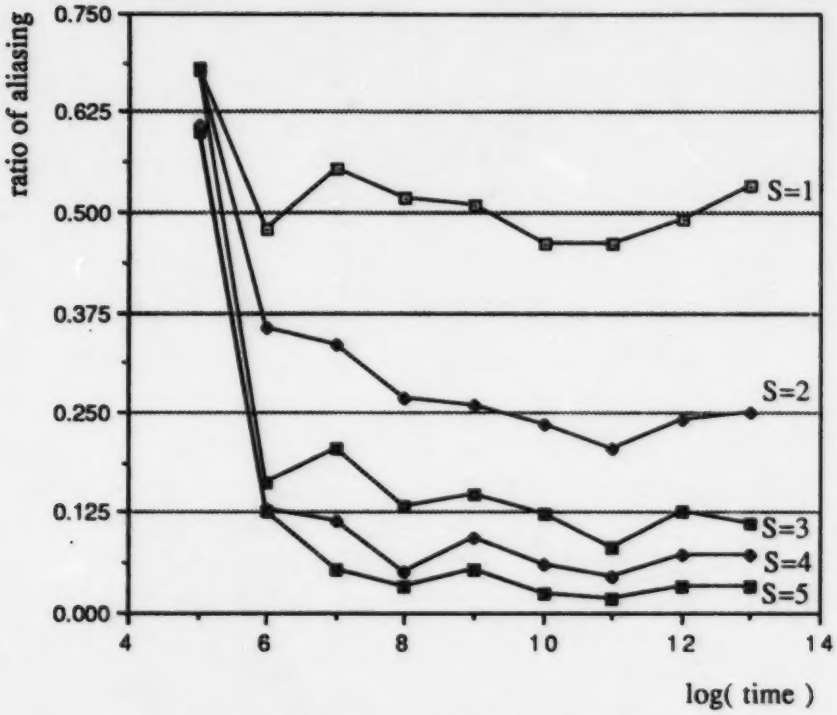

Fig. 15. Aliasing ratio for circuit c880.

The data presented in this section confirm that our conjecture about aliasing probability is fairly accurate. Therefore, in practice, one can estimate the aliasing probability using expression 2^{-S}. Another important observation is that, in practice, the probability of aliasing does not depend on the selection of the flip-flop from which the signature is collected.

6. FAULT COVERAGE ESTIMATION

As mentioned before, the quality of testing is often judged by fault coverage. In this section we will apply formulae (F.1), (F.2), and (F.3) to estimate the fault coverage for a circuit with CSTP.

Consider a combinational CUT and a set, FS, of its detectable faults. For each fault $f \in FS$, there exists a nonempty set, TS(f), of all test patterns that detect f. Fault coverage, FC, is defined as the fraction of faults in FS that are detected by a given testing procedure. Since any testing procedure is, in fact, a deterministic process - even when test patterns were generated randomly - the exact fault coverage for a

circuit can be determined by simulation. For a very large circuit, however, it may be very long and expensive. Hence, a method for fault coverage estimation would be very useful. Such method would be especially valuable at an early stage of circuit design. Of course, at an early stage one cannot give the exact number, but it should be possible to estimate the expected value. Expected fault coverage is defined as follows:

$$EFC = \frac{1}{|FS|} \sum_{f \in FS} pr(f \text{ is detected}), \tag{6.1}$$

where pr(event) denotes the probability of the event.

The probability that a fault, f, is detected depends on the probability that f is sensitized and the probability of aliasing. It can be expressed as

$$pr(\text{ f is detected }) = pr(\text{ f is sensitized }) \cdot [\ 1 - pr(\text{ aliasing })]. \tag{6.2}$$

Substituting $1-(1-q_f)^T$ for pr(f is sensitized) and (F.3) for pr(aliasing), we get

$$pr(\text{ f is detected }) = [1 - (1- q_f)^{T-L}] \cdot (1 - 2^{-S}).$$

This implies that, "effectively", $1 - (1- q_f)^{T-L}$ is the probability that f is sensitized, and 2^{-S} is the probability of aliasing. Using the inequality $L < P$, where P is the total number of flip-flops in the circular path, in the above equality, we obtain

$$pr(\text{ f is detected }) \geq [1 - (1- q_f)^{T-P}] \cdot (1 - 2^{-S}). \tag{6.3}$$

From (6.1) and (6.3), it follows that

$$EFC \geq \frac{1 - 2^{-S}}{|FS|} \sum_{f \in F} [1 - (1- q_f)^{T-P}]. \tag{6.4}$$

Consider a k-input combinational block *CL*. If the inputs of the CSTP section that generates test patterns for *CL* are stimulated by outputs of a block under test, formula (F.1) can be used to estimate the ESC. Note that the probability that a fault f is sensitized is equal to the probability that at least one of its test patterns, $t \in TS(f)$, is covered during a testing session. Moreover, for an arbitrary $t \in TS(f)$, the ESC can be used to estimate the probability that t is covered during a testing session. Thus, by (6.1) and (6.2), the EFC for *CL* can be estimated by ESC · [1 – pr(aliasing)], which in turn can be estimated by

$$[\,1-\exp(-\frac{T-P}{2^k})]\cdot(\,1-2^{-S}\,).$$

Of course, this estimation is very conservative, since usually there are many test patterns that detect a fault and the probability of generating at least one of them may be much larger than the probability that we have taken into account. In the case where a CSTP section is not stimulated by outputs of a block under test, there is no accurate formula to estimate the ESC. In this case, however, the probability that a fault is detected is not smaller than the probability of covering the hardest-to-generate test pattern. Thus, a lower bound on the EFC is given by

pr(the hardest to generate test pattern is covered) $\cdot$ $(1-2^{-S})$.

Since formula (F.2) provides a lower bound on the first factor of the above formula, it can also be used to estimate a lower bound on the EFC, i.e.,

$$[\,1-\exp(-\frac{T-P}{2^{k+1}})]\cdot(\,1-2^{-S}\,).$$

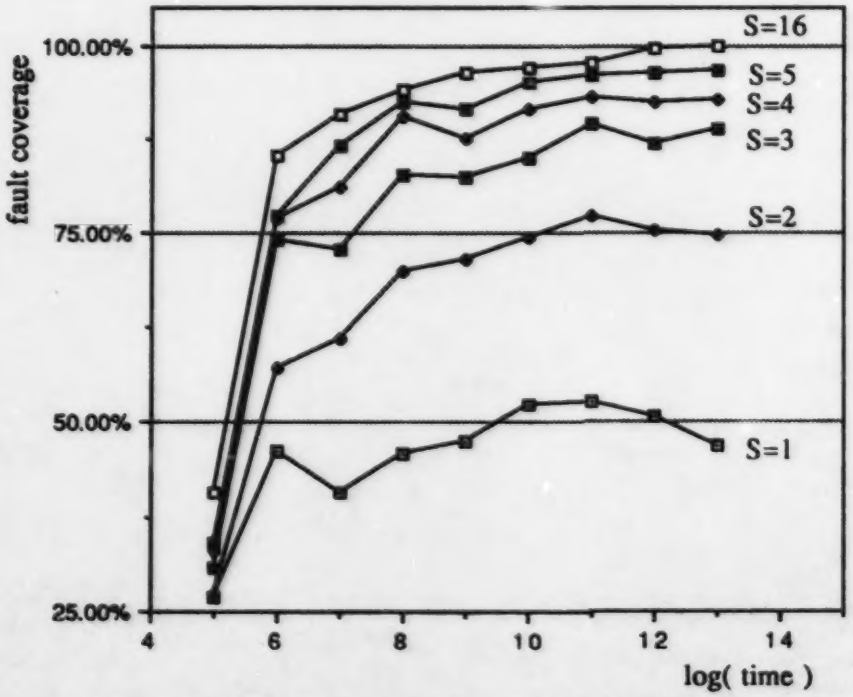

Fig. 16. Fault coverage for circuit c880.

As an example of the circular path fault detection effectiveness, consider circuit c880 that we used before to illustrate aliasing. Fig. 16 shows fault coverage as a

function of time for several signature lengths. It should be pointed out that all single stuck-at faults in this circuit are detectable, and the test set reported in [SCHAU89] consists of 70 test patterns. Assuming that a scan path [BMS87, WIPA82] is used to apply deterministically derived test patterns, and it contains only the input and output flip-flops (similarly to the circular path we used in our experiments), it would take at least $70 \cdot (60 + 1) + 28 = 4298$ clock cycles to test this circuit (in practice, probably many times more). Our simulation results showed that for testing time of 4096 clock cycles, CSTP offers fault coverage of 99.6%. For 8192 clock cycles, the fault coverage is 100%.

7. CONCLUSIONS

In this paper, we have studied the testing effectiveness of the BIST technique called Circular Self-Test Path. Its test generation effectiveness was evaluated by the expected state coverage and the probability that the hardest-to-generate test pattern is covered during a testing session. We developed formulae to estimate these values very accurately. These formulae show that test generation effectiveness is one minus a negative exponential function of time. The applicability of our formulae was verified through extensive simulation experiments on real circuits.

The aliasing aspect of CSTP was also analyzed, leading to a simple formula for aliasing probability estimation. The applicability of this formula was also verified through extensive simulation experiments, which indicate that it can be used in practice with very good accuracy. The formulae for test generation effectiveness and the formula for aliasing were used to develop formulae for expected fault coverage estimation. Although some of these formulae give a very pessimistic estimation when compared with the actual results for benchmark circuits, they indicate that for real circuits, testing time of a fraction of a second is enough to offer fault coverage very close to 100%, e.g, 99.999%. It is worth pointing out that the class of faults we have considered is not confined to single stuck-at faults; it consists of all detectable single and multiple faults that do not convert combinational blocks into sequential circuits The formulae can be used as tools by a designer who has to plan a testing session to satisfy given specifications for fault coverage. It is important that a testing session can be planned accurately without expensive detailed circuit simulation.

Finally, we can conclude that, in addition to the CSTP attributes listed in [KRPI89], CSTP is an effective BIST technique that offers very good testing quality

within a reasonable testing time (recall the comparison of fault coverage for deterministic testing and the CSTP-based testing of circuit c880 in the previous section). Thus CSTP proves to be an attractive alternative to the design-for-testability techniques developed previously.

8. APPENDIX

To prove that formula (F.2) constitutes a lower bound on the probability of covering the least probable state, let us start by recalling a statement from Section 3.3.3; namely, if v=1, then for any k and any input vector applied to the inputs of a CSTP section, the state transition graph for the section is isomorphic to the state transition graph for a shift register. Thus, if one proves that, for a shift register, state 0 is the least probable state, and the probability that state 0 is covered during a testing session is not lower than that given by formula (F.2), then one can conclude that if v = 1 then any state is covered with a probability not lower than that given by (F.2).

Consider the set $\Sigma(L)$ of all bit sequences of length L generated by a k-bit shift register. We are interested in computing the number of sequences in $\Sigma(L)$ that contain a given binary vector $V = v_1v_2 \ldots v_k$ as a subsequence. To do this, a finite automaton M_V, as introduced in the previous section, can be used. M_V has k+1 states, including the initial state and the final (or accepting) state. Each state corresponds to a prefix of V, including the null sequence. The final state corresponds to the entire V and remembers the fact that the input sequence received so far contains V. In general, the state corresponding to $v_1v_2...v_i$ (i<k) remembers that the input sequence received so far does not contain V, that the last i input symbols were $v_1v_2...v_i$, but the last i+1 input symbols were not $v_1v_2...v_{i+1}$. Therefore, both input symbols 0 and 1 take the final state back to itself. Similarly, for each state s, the transition out of s, in response to input symbol 0 or 1, can be easily determined.

As examples, two automata M_6 and M_0 are shown in Figs. 5 and 17, respectively, where the subscript is the numerical equivalent of the binary vector V.

Given automaton M_i, let $A_L(M_i)$ denote the set of all bit sequences of length L that are accepted by M_i. We want to show that $|A_L(M_0)| \leq |A_L(M_i)|$, for all i, $0 \leq i \leq 2^k-1$. Instead of proving this fact directly, we shall first transform M_i into another automaton M'_i such that $|A_L(M'_i)| = |A_L(M_i)|$, and then prove $|A_L(M_0)| \leq |A_L(M'_i)|$.

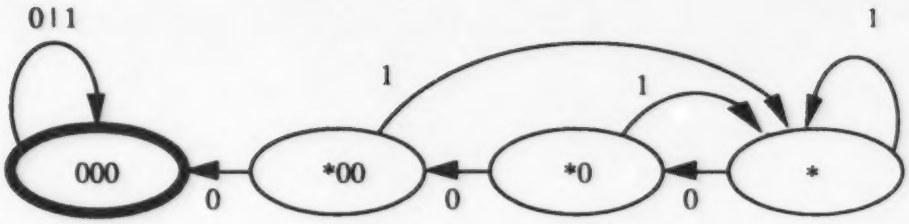

Fig. 17. Automaton M_0

Given M_i , M'_i is defined as follows:

1. state set of M'_i = state set of M_i

2. if $v_{i+1} = 1$, then interchange the labels of the two transitions out of the state corresponding to $v_1v_2 ...v_i$. (As illustrated in ," for i=0, this rule means that if the horizontal arrow out of a state is labeled by 1, then change it to 0 and change the label of the other transition out of this state to 1.)

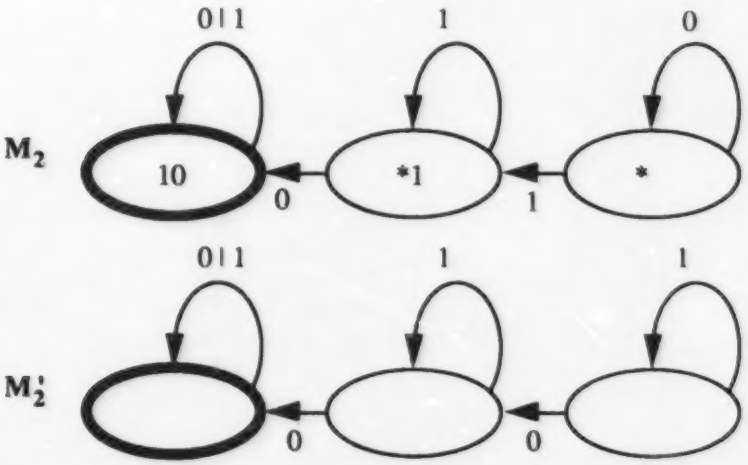

Fig. 18. Automaton transformation, $M_2 \rightarrow M'_2$

Fig. 18 shows such transformation for M_2 (k=2). Note that M_2 and M'_2 are the same except that the labels on transitions have been changed. It is easy to see that $A_3(M_2)$ = { 010, 110, 100, 101} and $A_3(M'_2)$ = {000, 100, 010, 001}.

Lemma 1: For any L and i ($0 \le i \le 2^k-1$), $|A_L(M'_i)| = |A_L(M_i)|$ holds.

Proof: Consider the graphical representations of M_i and M'_i. Note that any input sequence α for M_i uniquely defines a path $p(\alpha)$ in M_i starting in its initial state. Path $p(\alpha)$, in turn, uniquely defines an input sequence β in M'_i starting in its initial state. We make the following observations:

1. this mapping from α to β is one-to-one.
2. α belongs to $A_L(M_i)$ iff β belongs to $A_L(M'_i)$.

This completes the proof. □

Theorem 2:

For any L and i, $0 \leq i \leq 2^k-1$, $|A_L(M_0)| \leq |A_L(M_i)|$ holds.

Proof: From M_i, construct M'_i as above. By Lemma 1, it is sufficient to prove $|A_L(M_0)| \leq |A_L(M'_i)|$. Consider any $\alpha \in A_L(M_0)$. There are two cases to consider. Either α starts with k 0's or α contains a subsequence 1 followed by at least k 0's. In the first case, M'_i clearly accepts α as well. In the second case, consider the first such subsequence that appear in α. When the first 1 in that subsequence is input to M_0, M_0 is put in its initial state, and the k subsequent 0's take M_0 to its final state. On the other hand, if α is given to M'_i, when the first 1 in the subsequence arrives, it puts M'_i in a state which may or may not be its initial state. In any event, the k subsequent 0's take M'_i to its final state as well. □

Theorem 2 implies that state 0 is the least probable state. Thus, the next step in showing the validity of (F.2) is to prove that (F.2) expresses a lower bound on the probability that state 0 is covered during a testing session of T clock cycles.

As stated before, M_0 has k+1 states, s_0 (initial state), s_1, ..., s_k (final state), where s_i ($i \neq k$), labeled by $*v_1 v_2 \ldots v_i$, remembers the fact that the i most recent input symbols were all 0's, but the i+1 most recent input symbol was not a 0. With each state s_i of M_0, we associate a number $N(s_i, L)$, which denotes the number of *bit sequences* of length L that take M_0 from state s_0 to s_i. In order to compute $N(s_i, L)$ for L = 1, 2, ..., we first define

$$N(s_0, 0) = 1$$

$$N(s_i, 0) = 0 \text{ for all } i,\ 1 \leq i \leq k. \qquad (8.1)$$

Now observe that, by definition, $N(s_0, 1) = 1$ and $N(s_1, 1) = 1$. Note that, in general,

$$N(s_i, L) = N(s_{i-1}, L-1) \text{ for all } i,\ 1 \le i \le k-1, \tag{8.2}$$

since each sequence contributing to $N(s_i, L)$ is obtained by appending a 0 at the end of a sequence counted in $N(s_{i-1}, L-1)$, provided that $N(s_{i-1}, L-1) > 0$. As for $i = 0$, we have

$$N(s_0, L) = N(s_0, L-1) + N(s_1, L-1) + \ldots + N(s_{k-1}, L-1), \tag{8.3}$$

since each sequence contributing to $N(s_0, L)$ is obtained by appending a 1 at the end of a sequence counted in the expression on the right hand side. Using (8.2) repeatedly, (8.3) can be transformed into

$$N(s_0\ L) = N(s_0, L-1) + N(s_0, L-2) + \ldots + N(s_0, L-k). \tag{8.4}$$

For convenience, we define $f(T) = N(s_0, T+k)$. By (8.3), $f(T)$ is the number of bit sequences of length $T+k-1$ which take M_0 from s_0 to a state other than the accepting state. From (8.1) and (8.4), sequence $f(T)$, $T = -k, -k+1, \ldots, 0, 1, 2, \ldots$ is defined as follows:

$$\begin{cases} f(T) = 0 \text{ for } T < -k \\ f(-k) = 1 \\ f(T) = \displaystyle\sum_{i=1}^{k} f(T-i) \text{ for } T \ge -k+1. \end{cases} \tag{8.5}$$

Lemma 2: $f(T) < (2 - 1/2^k)\, f(T-1)$ for all $T \ge 1$. (8.6)

Proof: Basis: We must show that (8.6) is satisfied for all T, $1 \le T \le k$.

From the definition of $f(T)$ given above, it is not difficult to derive the following formulae:

$$\begin{cases} f(T) = 2^{k+T-1} & \text{for all T, } -k+1 \le T \le 0. \\ f(T) = 2^{k+T-1} - (T+1)2^{T-2} & \text{for all T, } 1 \le T \le k. \end{cases} \tag{8.7}$$

Therefore, we have

$$(2 - 1/2^k)\, f(T-1) = 2^{k+T-1} - (T+1)2^{T-2} + T/2^{k-T+3} \quad \text{for all } T,\ 2 \le T-1 < k. \qquad (8.8)$$

By subtracting (8.7) from (8.8), we obtain

$$(2 - 1/2^k)\, f(T-1) - f(T) = T/2^{k-T+3} > 0 \quad \text{for all } T,\ 2 \le T < k+1.$$

As for T = 1, it is easy to show that $f(1) < (1-1/2^k)\, f(0)$, since $f(1) = 2^k - 1$ and $f(0) = 2^{k-1}$.

Induction hypothesis: (8.6) is true for all T–1, T–2, ..., T–k for some $T \ge k+1$.

From the hypothesis, it follows that

$$f(T) = \sum_{i=1}^{k} f(T-i) < (2 - 1/2^k)\, [\, \sum_{i=2}^{k+1} f(T-i)\,] = (2 - 1/2^k)\, f(T-1). \square$$

Theorem 3: $f(T) < 2^{k+T-1}(1 - 1/2^{k+1})^T$ for all $T \ge 1$.

Proof: Solving the recurrence inequality (8.6), we obtain

$$f(T) < (2 - 1/2^k)^{T-1} f(1).$$

Since $f(1) = 2^k - 1$ by (8.7), we have

$$f(T) < (2^k - 1)2^{T-1}(1 - 1/2^{k+1})^{T-1} < 2^{k+T-1}(1 - 1/2^{k+1})^T. \square$$

Because a k-bit shift register has 2^{k+T-1} different and equally likely state sequences of length T, the fraction of the state sequences of length T that contain vector 0 is given by $1 - \frac{f(T)}{2^{k+T-1}}$. Formula (F.2) now follows from Theorem 3 (recall the derivation of (F.1)).

REFERENCES

[BAMC82] P. H. Bardell and W. H. McAnney, "Self-Testing of Multichip Logic Modules," Proc. IEEE Int. Test Conference, pp. 200-204, 1982.

[BMS87] P. H. Bardell, W. H. McAnney, and J. Savir, "Built-In Test for VLSI: Pseudorandom techniques," New York: John Wiley & Sons, 1987.

[BRFU85] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in FORTRAN," Proc. IEEE Int. Symp. Circuits and Systems; (Special Session on ATPG and Fault Simulation), June 1985.

[DAV90] R. David, "Comments on 'Signature Analysis for Multiple Output Circuits," IEEE Trans. Comp., pp. 287-288, Feb. 1990.

[DOFER89] M. Damiani, P. OLivo, M. Favalli, S. Ercolani, and B. Ricco, "Aliasing in Signature Analysis Testing with Multiple-Input Shift Registers," Proc. European Test Conference, pp. 346-353, Apr. 1989.

[DOFR89] M. Damiani, P. OLivo, M. Favalli, and B. Ricco, "An Analytical Model for the Aliasing Probability in Signature Analysis Testing," IEEE Trans. Computer-Aided Design, pp. 1133-1144, Nov. 1989.

[DTM90] "The Challenges of Self-Test," IEEE Design & Test of Computers, pp. 46-56, February 1990.

[HUPE87] C. Hudson and G. Peterson, "Parallel Self-Test with Pseudo-Random Test Patterns," Proc. IEEE Int. Test Conference, pp. 954-963, 1987.

[IVAG89] A. Ivanov and V. K. Agarwal, "An Analysis of the probabilistic behavior of Linear Feedback Signature Registers," IEEE Trans. Computer-Aided Design, pp. 1074-1088, Oct. 1989.

[KMZ79] B. Konemann, T. Mucha, and G. Zwiehoff, "Built-In Logic Block Observation Techniques," Proc. IEEE Int. Test Conference, pp. 37-41, 1979.

[KRPI87] A. Krasniewski, and S. Pilarski, "Circular Self-Test Path: A Low Cost BIST Technique for VLSI Circuits," Proc. ACM/IEEE Design Automation Conference, pp. 407-4015, 1987.

[KRPI89] A. Krasniewski, and S. Pilarski, "Circular Self-Test Path: A Low Cost BIST Technique for VLSI Circuits," IEEE Trans. Computer-Aided Design, pp. 46-55, January 1989.

[MCCL89] "McCluskey's dictionary of common testability terms," IEEE Design & Test of Computers, p. 72, June 1989.

[MIZH89] D. M. Miller and S. Zhang, "Aliasing in Multiple-Input Data Compactors," Proc. Canadian Conf. on Elect. and Comp Eng., pp. 347-351, Sept. 1989.

[PIKR88] S. Pilarski, and A. Krasniewski, "Effectiveness of Fault Detection for Low Overhead Self-Testing VLSI Circuits," Proc. IEEE Int. Symp. Circuits and Systems, pp. 451-454, 1988.

[POLB88] M. M. Pradhan, E. J. O'Brien, S. L. Lam, and J. Beausang, "Circular BIST with Partial Scan," Proc. IEEE Int. Test Conference, pp. 719-729, 1988.

[SABA84] J. Savir and P.Bardell, "On Random Test Length," IEEE Trans. Comp., pp. 467-474, June 1984.

[SCHAU89] M. H. Schulz, and E. Auth, "Improved Deterministic Test Pattern Generation with Applications to Redundancy Identification," IEEE Trans. Computer-Aided Design, pp. 811-816, July 1989.

[STRO88a] C. E. Stroud, "An Automated BIST Approach for General Sequential Logic Synthesis," Proc. ACM/IEEE Design Automation Conference, pp. 3-8, 1988.

[STRO88b] C. E. Stroud, "Automated BIST for Sequential Logic Synthesis," IEEE Design & Test of Computers, pp. 22-32, Dec. 1988.

[VAB84] P. Varma, A. P. Ambler, and K. Baker, "An Analysis of the Economics of Self-Test," Proc. IEEE Int. Test Conference, pp. 20-30, 1984.

[WDGS88] T. W. Williams, W. Daehn, M. Gruetzner, and C. W. Starke, "Bounds and Analysis of Aliasing Errors in Linear Feedback Shift Registers," IEEE Trans. Computer-Aided Design, pp. 75-83, Jan. 1988.

[WIDA89] T. W. Williams and W. Daehn, "Aliasing Errors in Multiple Input Signature Analysis Registers," Proc. European Test Conference, pp. 388-345, Apr. 1989.

[WIPA82] T. W. Williams and K. P. Parker, "Design For Testability - A Survey," IEEE Trans. Comp., pp. 2-15, Jan. 1982.

Centre for Systems Science

• Simon Fraser University established the **Centre for Systems Science** as a provincial centre for advanced study. The objective of CSS is twofold: coordinate and combine advanced systems research at SFU, and encourage application of research results within industry in order to further the technological development of British Columbia.

• CSS has assembled three research groups to streamline its efforts. The orientation of these groups is as follows: intelligent systems, microelectronics, and computer and communications systems. Because of shared research interests, this latter group merges with LCCR (**Laboratory for Computer and Communications Research**) for the publication of technical reports.

• CSS's three groups involve over 80 skilled faculty researchers principally from SFU's Schools of Engineering Science and Computing Science, but also draw substantially on Mathematics, Physics, Geography, Philosophy, Linguistics, Psychology, Communication, and Kinesiology.